IDENTITYMARK: SOURCE-SIDE IDENTITY WATERMARKING FOR DEEPFAKE PROVENANCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Face-swapping deepfakes can misuse publicly shared portraits to transfer a person's likeness onto attacker-selected target images, spreading misleading information or damaging the victim's reputation. Since the resulting deepfake may retain little or none of the source portrait's pixels or shape, conventional watermarks become ineffective in this scenario. We introduce IdentityMark, a source-side identity watermarking framework that traces such deepfakes to the protected sources, supporting deepfake verification when the target image is beyond the defender's control. Rather than embedding provenance into image content, IdentityMark encodes a binary message into the facial identity representation transferred during face swapping. A message-to-identity autoencoder produces a message-bearing identity embedding, while diffusion-based latent optimization moves the source identity embedding toward this target while preserving visual fidelity and the original identity. Given a suspicious face-swapped image, the message is decoded directly from its facial identity embedding and compared against the victim's registered watermarks to verify its authenticity. IdentityMark requires no access to or training through downstream face-swap models, making it agnostic to the face-swapping method. In our evaluated face-swapping setting, IdentityMark achieves 71.66% bit accuracy, compared with approximately 50% for other watermarking methods, consistent with random guessing. These results demonstrate the feasibility of source-side provenance protection for tracing and verifying face-swapping deepfakes.

1 INTRODUCTION

Face swapping takes two images with distinct roles, as illustrated in Fig. 1. The *source* portrait provides the facial identity, while the *target* image determines the pose, expression, illumination, and spatial context of the output. In realistic misuse, these inputs may be controlled by different parties: a person can protect a published portrait, while an attacker can later choose an arbitrary target image and face-swapping model. This asymmetry motivates a source-side provenance problem: can a published portrait be protected such that, when its identity is transferred into a manipulated image, the output retains a recoverable provenance signal linking it to the registered source? Such a link can support identity-reuse attribution and retrieval of a trusted source record for verification.

This setting is particularly challenging for conventional image watermarking because face swapping combines multiple transformations that disrupt watermark-carrying structures. While modern methods can withstand individual degradations such as compression, resizing, cropping, perspective changes, or localized editing (Tancik et al., 2020; Bui et al., 2025; Sander et al., 2025; Zhang et al., 2024), face swapping can jointly alter pixel values, scale, shape, viewpoint, pose, expression, and local appearance. The source face is effectively re-rendered under the geometry and attributes of a different target, substantially disrupting watermark signals encoded in spatial, frequency, texture, or other image-dependent structures. Thus, the challenge lies not in a single severe degradation, but in the composition of transformations that jointly destroy the original watermark-carrying channels. Further literature review is given in Appendix. A.

Face swapping is explicitly designed to preserve the source *identity*. Modern systems commonly use identity encoders, conditioning modules, or preservation objectives to maintain the source subject's identity Chen et al. (2020); Rosberg et al. (2022); Zhao et al. (2023). In contrast, non-identity attributes such as pose, expression, and viewpoint are largely inherited from the target image Xu et al.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

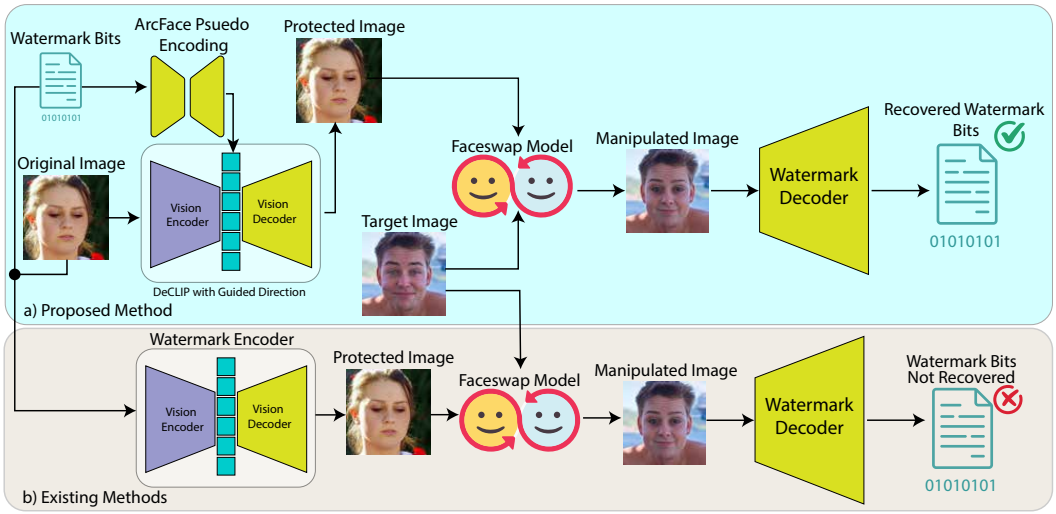

Figure 1: Face-swapping protocol and protection locations. The source image provides the identity to be transferred, whereas the target image provides most pose, expression, background, and other visual attributes retained in the output. Prior proactive watermarking methods commonly protect the target/carrier image, while our setting embeds provenance into the identity-providing source itself.

(2022); Chen et al. (2020). Identity therefore provides a natural semantic carrier for provenance. We embed the watermark into the transferable identity representation so that it follows the information path preserved by the face-swapping model.

To this end, we introduce IdentityMark, a source-side identity watermarking framework that encodes a binary provenance message into transferable facial identity. Given a source portrait I_s and a message w, a message-to-identity autoencoder constructs a message-bearing target in face-embedding space. Diffusion-based latent optimization (Ho et al., 2020; Kim et al., 2022) then produces a protected portrait $\hat{I}_s$ whose appearance and perceived identity remain close to the original while its facial embedding is shifted toward the target. When an unknown downstream model transfers this identity to an arbitrary target image, the manipulated face retains part of the message-bearing signal, which a decoder recovers from its facial identity embedding. Neither watermark learning nor source protection requires access to, differentiation through, or training on the downstream face-swapping model.

The key challenge is to encode a signal that survives downstream identity extraction and synthesis without compromising the protected portrait's visual identity or usability. In our evaluated source-side setting, IdentityMark achieves approximately 71.66% bit accuracy, whereas conventional image-watermarking baselines remain near 50%, indicating that facial identity can serve as a more persistent provenance carrier than source-image content under face swapping.

Our contributions are summarized as follows:

- We formulate *source-side identity provenance* for face swapping, where the defender controls only the identity-providing source portrait while the attacker can choose arbitrary target and swapping model. A recovered message can link a manipulated face to a registered protected source, enabling provenance attribution and follow-up identity verification.
- We introduce IdentityMark, which maps binary provenance messages into face-identity space and realizes the resulting message-bearing identity in a source portrait through diffusion-based latent optimization, allowing the provenance signal to follow identity transfer rather than persistent source-image pixels.
- We show that the proposed provenance signal transfers across downstream face-swapping models without requiring access to or training through those models, in average achieving approximately 71.66% bit accuracy in our evaluated source-side setting while conventional image-watermarking baselines remain near random guessing.

2 Problem Formulation and Threat Model

2.1 Source Registration and Protection

We consider a social-media deployment in which users publicly share personal portraits through an online platform. Before a portrait I_s is published, the platform automatically assigns a provenance message $w \in \{0,1\}^k$ and applies the protection algorithm P to obtain

$$\hat{I}_s = P(I_s, w). \tag{1}$$

The protected portrait $\hat{I}_s$ is then published as the user's ordinary social-media image, while the platform stores the corresponding registered message w for later verification.

The protected portrait should remain perceptually similar to the original while its machine-readable face representation is shifted toward the watermark-conditioned pseudo-identity. We therefore require

$$d_{\mathrm{per}}(\hat{I}_s, I_s) \le \epsilon, \tag{2}$$

while encouraging

$$E(\hat{I}_s) \approx e_w, \qquad e_w = H(w), \tag{3}$$

where d_{per} denotes a perceptual distance, H maps the provenance message to a pseudo-identity embedding, and E is the face encoder used to realize the watermark in identity space.

The embedded provenance signal is intended to remain recoverable under common image degradations and, more importantly, after downstream face-swapping operations.

2.2 Attack and Manipulation

Once the protected portrait is publicly available on the social-media platform, an attacker may download $\hat{I}_s$ and use it as the identity-providing source for an arbitrary face-swapping model G. Given an attacker-selected target image I_t, the manipulated output is

$$Y = G(I_t, \hat{I}_s). \tag{4}$$

The resulting image may then be redistributed on the same platform or elsewhere for impersonation, defamation, or other misuse.

2.3 Provenance Verification

When a suspicious image Y later appears on the platform, a platform-side verification service may analyze it using conventional passive deepfake detectors. Our method provides an additional source-side provenance signal.

In targeted misinformation or reputation attacks, the manipulated image must reach an audience that recognizes, or is informed of who the victim is; otherwise, the content cannot meaningfully affect that person's reputation. We therefore assume that the claimed victim is known at verification time and compare only against watermarks registered for that victim, rather than searching over all platform users.

The verifier extracts a face representation $f(Y)$ and recovers

$$\hat{w} = D(f(Y)). \tag{5}$$

If $\hat{w}$ matches a watermark registered for the claimed victim, this provides evidence that the corresponding protected portrait was involved in the manipulation. The recovered provenance therefore complements passive real/fake detection with source-level attribution.

3 Method

3.1 Overview

IdentityMark consists of three components. As shown in Fig. 2, first, we learn a *message-to-identity* mapping that converts a binary watermark w into a pseudo facial-identity embedding p_w, together

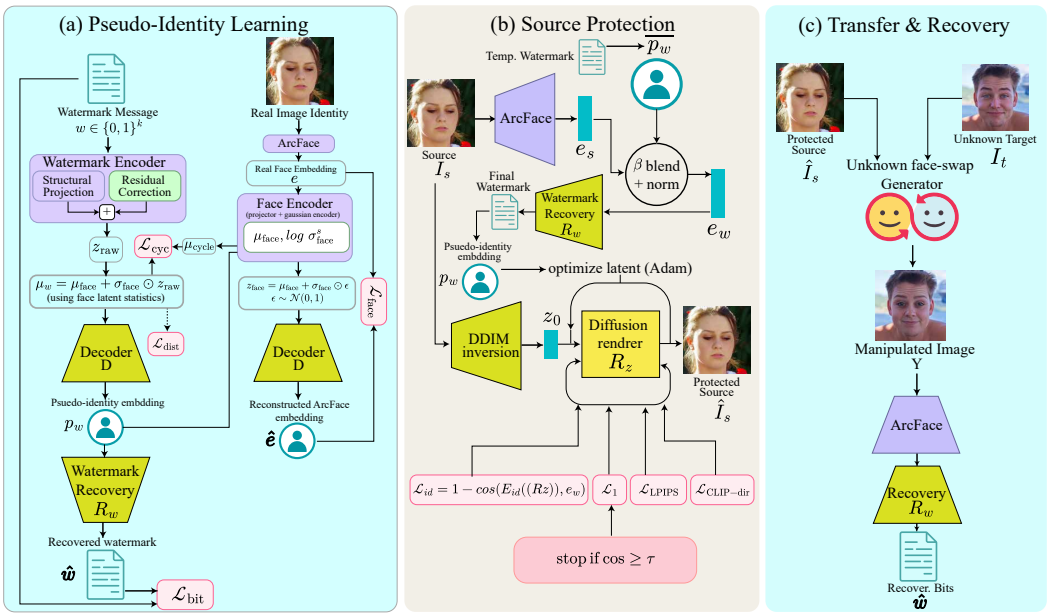

Figure 2: Overview of IdentityMark. A binary provenance message is mapped to a pseudo-identity embedding, realized in a source portrait through image optimization, transferred by a downstream face-swapping model, and finally recovered from the identity embedding of the manipulated output.

with a recovery network that maps identity embeddings back to watermark bits. Second, given a source portrait I_s, we optimize the image such that the protected portrait $\hat{I}_s$ remains perceptually similar to I_s, while its machine-readable face embedding is shifted from the original identity toward p_w. Third, when a protected identity later appears in a face-swapped or otherwise manipulated image, we extract its face embedding and recover the watermark using the same recovery network.

Importantly, neither the message-to-identity model nor the watermark recovery network is trained through a downstream face-swapping model.

3.2 Message-to-Identity Construction

Let $w \in \{0,1\}^{d_w}$ denote a binary provenance message and $e \in \mathbb{S}^{d_e-1}$ a normalized face embedding extracted by ArcFace Deng et al. (2019). We learn a message-to-identity autoencoder that maps each watermark to a decodable pseudo-identity in the facial embedding space.

The watermark bits are first converted to a bipolar representation. The watermark encoder combines a structured linear code with a small nonlinear residual correction. P_{str} is orthogonally initialized and regularized to preserve approximately independent watermark directions.

$$z_{\text{raw}} = \alpha P_{\text{str}}(w) + \lambda_{\text{res}} R_w(P_w(w)), \qquad \mu_w = \mu_{\text{face}} + \sigma_{\text{face}} \odot z_{\text{raw}}, \tag{6}$$

where μ_{face} and s_{face} are the per-dimension mean and scale of real-face latents. This anchors each watermark center μ_w to the latent space learned from real face embeddings.

To provide tolerance to identity shifts that may occur after face swapping, training samples a local neighborhood around each watermark center, using a fixed noise scale σ_w

$$z_w = \mu_w + \sigma_w \sigma_{\text{face}} \odot \epsilon, \qquad \epsilon \sim \mathcal{N}(0, I). \tag{7}$$

The sampled latent is decoded into a normalized pseudo-identity embedding,

$$e_w = \text{norm}(D(z_w)). \tag{8}$$

At inference time, we use the deterministic center $z_w = \mu_w$ to obtain the pseudo-identity associated with a watermark.

A watermark recovery head R_w is jointly learned such that

$$\hat{w} = R_w(e_w). \tag{9}$$

The same latent decoder is also trained to reconstruct real face embeddings, encouraging pseudo identities to remain compatible with the representation learned from real facial identities. The overall training objective is

$$\mathcal{L}_{\mathrm{AE}} = \lambda_{\mathrm{face}}\mathcal{L}_{\mathrm{face}} + \lambda_{\mathrm{bit}}\mathcal{L}_{\mathrm{bit}} + \lambda_{\mathrm{cyc}}\mathcal{L}_{\mathrm{cyc}} + \lambda_{\mathrm{dist}}\mathcal{L}_{\mathrm{dist}} \tag{10}$$

Here, $\mathcal{L}_{\mathrm{face}}$ preserves reconstruction of real face embeddings, $\mathcal{L}_{\mathrm{bit}}$ enforces watermark recovery with additional emphasis on difficult bits, $\mathcal{L}_{\mathrm{cyc}}$ stabilizes the watermark latent representation, and $\mathcal{L}_{\mathrm{dist}}$ encourages watermark latents to follow the statistics of real face latents.

At inference time, we use the deterministic center of the learned watermark representation rather than adding random noise. The resulting decoded pseudo identity is used directly as the target embedding for image realization.

3.3 Protected-Image Optimization

Given a source portrait I_s and its normalized pseudo-identity target p_w, our second stage realizes the watermark in image space. The objective is deliberately asymmetric: the protected portrait should remain visually similar to the original person, while its machine-readable facial identity is moved toward the watermark-derived pseudo identity.

Let $\hat{I}_s = R(z)$ denote the image rendered from an optimizable diffusion latent z. We optimize

$$\mathcal{L}_{\mathrm{img}} = \lambda_{\mathrm{id}}\mathcal{L}_{\mathrm{id}} + \lambda_{l1}\mathcal{L}_1 + \lambda_{\mathrm{lpips}}\mathcal{L}_{\mathrm{lpips}}, \tag{11}$$

where

$$\mathcal{L}_{\mathrm{id}} = 1 - \cos\left(E_{\mathrm{id}}(\hat{I}_s), p_w\right), \tag{12}$$

and $\mathcal{L}_1$ and $\mathcal{L}_{\mathrm{lpips}}$ constrain the pixel-level and perceptual deviation from the original portrait.

Thus, visual similarity and embedding similarity play different roles: I_s acts as the perceptual reference, whereas the other p_w acts as the machine-identity target. The optimization therefore seeks a protected portrait that remains recognizable as the original subject to a human observer while being represented by the face encoder near the watermark-derived pseudo identity. Algorithm 1 shows a detailed process of image optimization.

Algorithm 1 Protected Image Optimization

Require: Source portrait I_s, provenance message w, message-to-identity module F_w, face encoder E_{id}, maximum iterations N, identity threshold τ, watermark encoder E, pseudo-generator D

Ensure: Protected portrait $\hat{I}_s$

$\bar{p}_w \leftarrow F_w(w)$
$e_s \leftarrow E_{\mathrm{id}}(I_s)$
$p_w \leftarrow \mathrm{norm}(D(E(\mathrm{norm}((1-\alpha)e_s + \alpha\bar{p}_w))))$
$z_0 \leftarrow \mathrm{DDIMInv}(I_s)$
$z \leftarrow z_0$
for $t = 1, \ldots, N$ **do**
  $I_t \leftarrow \mathcal{D}DIM(z; z_0)$
  $e_t \leftarrow E_{\mathrm{id}}(I_t)$
  $\mathcal{L} \leftarrow \mathcal{L}_{\mathrm{img}}(I_t, I_s, e_t, p_w)$
  $z \leftarrow \mathrm{AdamStep}(z, \nabla_z\mathcal{L})$
  if $\cos(e_t, p_w) \geq \tau$ **then**
    break
  end if
end for
$\tilde{I}_s \leftarrow \mathcal{D}DIM(z; z_0)$
$\hat{I}_s \leftarrow \mathrm{QualityRefine}(\tilde{I}_s, I_s, p_w)$
return $\hat{I}_s$

Quality refinement. After identity realization, we perform a lightweight refinement that removes unnecessary image residuals while maintaining both proximity to the pseudo-identity target and recoverability of the embedded message. The refinement jointly constrains identity transfer, watermark recovery, and visual fidelity. Detailed refinement terms and hyperparameters are provided in the Appendix.

3.4 Watermark Recovery

Given a suspicious image Y, we detect and align its face, extract its normalized identity embedding, and recover the watermark using the recovery head trained in the message-to-identity stage:

$$e_Y = E_{\mathrm{id}}(Y), \qquad \ell = R_w(e_Y), \qquad \hat{w}_j = \mathbf{1}[\sigma(\ell_j) \geq 0.5]\,. \tag{13}$$

No face-swapping model is involved in recovery training. Robustness to face swapping therefore arises from encoding provenance in the facial identity representation that downstream swapping systems are designed to transfer, rather than from explicitly training against a particular face-swap pipeline.

Table 1: Bit accuracy (BA, %) after manipulation with different face-swap and cheapfake methods.

Method	SepMark-T (128 b)	EditGuard (64 b)	StableSignature (48 b)	LIDMark (152 b)	Ours (32 b)
SimSwap Chen et al. (2020)	50.22	49.81	49.75	49.85	**75.23**
InSwapper Wang (2023)	50.18	49.67	49.62	50.66	**77.38**
InStyleSwapper VisoMaster Team (2025)	49.99	49.83	48.75	49.67	**77.02**
Reface Reface (2020)	49.98	50.45	49.69	50.64	**79.37**
FaceDancer Rosberg et al. (2022)	50.08	50.22	50.32	47.38	**56.33**
DiffSwap Zhao et al. (2023)	49.27	50.03	47.95	48.98	**60.31**
UniFace Xu et al. (2022)	49.88	50.13	48.22	48.27	**74.78**
HyperSwap FaceFusion (2025)	50.28	50.86	50.06	48.14	**75.59**
2D Cheapfake Canu (2019)	50.31	49.63	48.88	46.50	**72.49**
3D Cheapfake Kowalski (2016)	50.22	49.71	49.51	48.98	**68.06**

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Data. We use CelebA-HQ following the predefined split provided by the official DiffusionCLIP Kim et al. (2022) repository, with 29,000 images for training and 500 images for testing. All images are processed at 256×256 resolution.

Pseudo-Identity Learning. The model is trained for 200 epochs on the face embedding from 29,000 CelebA-HQ training images, with random 32-bit watermarks sampled online. We use $\lambda_{\text{face}} = 1$, $\lambda_{\text{bit}} = 4$, $\lambda_{\text{cyc}} = 1$, and $\lambda_{\text{dist}} = 1$ for the four main objective terms, with the structured-code orthogonality and radius regularizers both weighted by 1.

Source Protection. For each source image, the DiffusionCLIP optimization is run for at most 150 iterations. Each optimization iteration uses 6 denoising steps, and early stopping is applied once the cosine similarity between the output face embedding and the target pseudo embedding reaches 0.95. The optimization objective combines perceptual quality and pseudo identity similarity with weights $\lambda_{\text{id}} = 1.0$, $\lambda_{\ell_1} = 1.0$ and $\lambda_{\text{lpips}} = 1.0$

Evaluation. All evaluations are conducted on the 500 CelebA-HQ test images. We evaluate both robustness to conventional image degradations and robustness to face-swapping methods. Watermark recovery is measured using bit accuracy (BA), while image fidelity between the original and protected images is evaluated using PSNR, SSIM, and LPIPS.

4.2 ROBUSTNESS TESTING

The central question of our evaluation is whether information embedded into a source portrait remains recoverable after its identity is transferred into a different target image. Table 1 shows that our method consistently retains substantially higher bit accuracy across face-swapping and cheapfake pipelines, while the evaluated baselines remain close to chance-level agreement. This **does not** imply that these methods are inherently less effective; rather, they are designed to solve other problems.

Table 2 further shows that watermark recovery is related to how strongly the manipulated output preserves the source identity. Since IdentityMark carries provenance through the source identity representation, weaker source–faceswapped identity similarity generally leaves less identity-carried watermark information available for recovery.

We further evaluate robustness under common image transformations, including JPEG compression, resizing, Gaussian blur, additive pixel noise, brightness and contrast changes, and center cropping. As shown in Table 3, our method achieves 99.55% clean bit accuracy and an average of 98.20% across all 20 attacked settings, with particularly strong robustness to center cropping compared with the evaluated baselines.

Moreover, as a sanity check, applying the watermark decoder to unprotected images yields a bit accuracy of 50.74%, close to the random binary watermarks.

Table 2: Identity cosine similarity between the face-swapped output and the protected source or target across manipulation methods. Higher source–output similarity indicates stronger source-identity preservation, while higher target–output similarity indicates greater target-identity retention.

Method	Source – output	Target – output
SimSwap	0.471	0.190
InSwapper	0.884	0.062
InStyleSwapper	0.803	0.098
Reface	0.318	0.103
FaceDancer	0.096	0.467
DiffSwap	0.253	0.444
UniFace	0.283	0.232
HyperSwap	0.767	0.137
2D Cheapfake	0.430	0.126
3D Cheapfake	0.323	0.291

Table 3: Bit accuracy (%) under common image degradations. The average is computed over settings.

Attack		SepMark-T (128 b)	Stable Signature (48 b)	EditGuard (64 b)	LIDMark (152 b)	Ours (32 b)
Protected		100.00	99.76	99.98	100.00	99.55
JPEG	Q90	100.00	93.17	98.35	100.00	99.27
	Q70	100.00	86.14	92.10	99.01	99.04
	Q50	100.00	80.23	86.00	95.19	98.87
	Q30	99.70	73.16	75.75	83.83	97.50
Resize	0.75	100.00	88.29	98.10	100.00	99.34
	0.50	100.00	73.26	6.34	100.00	98.57
	0.25	97.30	64.53	33.60	99.99	94.48
Blur	0.5	100.00	97.09	99.96	100.00	99.17
	1.0	100.00	71.93	70.53	100.00	97.92
	2.0	100.00	62.18	20.04	100.00	91.51
Noise	2	100.00	99.76	100.00	100.00	99.39
	5	100.00	98.99	99.85	99.95	98.93
	10	100.00	64.82	96.23	97.88	97.77
Brightness	0.8	100.00	98.66	99.99	100.00	99.31
	1.2	100.00	99.93	99.98	99.57	99.05
Contrast	0.8	100.00	98.55	99.98	100.00	99.27
	1.2	100.00	99.88	99.95	99.67	99.01
Center Crop	0.90	50.60	92.00	50.41	48.01	98.38
	0.75	50.00	75.84	50.40	49.60	98.19
VAE		99.99	49.72	53.24	99.94	98.93
Avg.		95.12	84.85	78.66	94.24	**98.33**

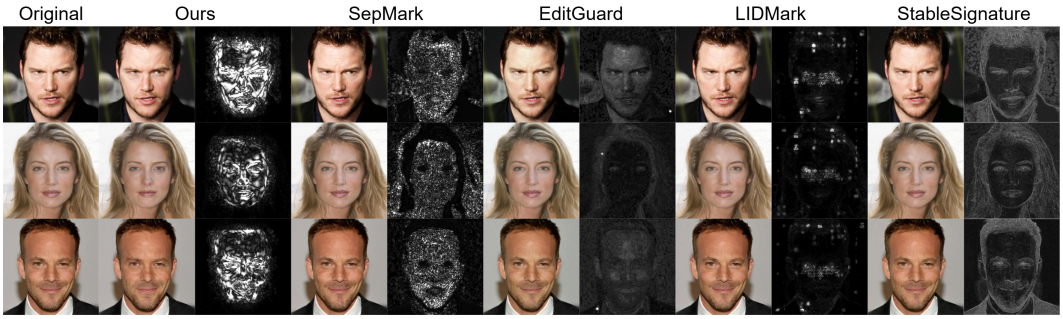

Figure 3: Qualitative comparison of protected source images produced by different watermarking methods. Absolute pixel differences from the original are shown at 20× amplification for visibility.

4.3 Visual Quality

We evaluate the visual fidelity of the protected source images in Table 4. The 32-bit configuration achieves ≈ 31 dB PSNR, with the distortion arising from small pixel-level changes induced by subtle modifications to facial characteristics. We additionally provide qualitative comparisons in Figure 3, showing that the protected images remain visually close to the originals, with the differences arising primarily from subtle changes in facial features rather than pixel-level perturbations.

Table 4: Visual quality of the protected source images.

Method	PSNR ↑	SSIM ↑	LPIPS ↓
Ours (32 b)	31.96	0.94	0.0156
SepMark-T (128 b)	38.52	0.97	0.0078
StableSignature (48 b)	37.71	0.94	0.0493
EditGuard (64 b)	41.80	0.99	0.0057
LIDMark (152 b)	43.12	0.99	0.0064

Despite this visual similarity, the average identity cosine similarity between the original and protected images is only 0.24, indicating a substantial shift in the identity representation.

4.4 Effect of Payload Capacity

We finally study how the proposed protection scales with payload length. While the main experiments use a 32-bit message, we additionally evaluate 64-bit and 128-bit payloads using the same source identities and evaluation protocol. Increasing the payload requires the protection procedure to encode more information while preserving source identity and visual fidelity. We therefore compare clean

Table 5: Payload-length ablation for 64-bit and 128-bit source-identity watermarks. For each degradation category, we report the average bit accuracy (%) over all tested severity levels within that category.

Method	Unprotected	Protected	JPEG	Resize	Blur	Noise	Brightness	Contrast	Cen. Crop	SimSwap	InSwapper
Ours (64 b)	51.16	99.17	97.38	95.56	94.21	97.71	98.16	98.22	97.24	70.78	72.65
Ours (128 b)	54.32	99.08	96.24	94.55	92.85	96.53	97.83	97.87	95.94	68.02	69.76

recovery, recovery after face swapping, and image quality across payload sizes to characterize the capacity–robustness trade-off.

The 64-bit configuration achieves a PSNR of 31.30 dB and an SSIM of 0.9302, while the 128-bit configuration achieves a PSNR of 33.72 dB and an SSIM of 0.9588. These values indicate that increasing the payload length does not necessarily lead to greater visual distortion in the current configuration, although the effect on watermark recoverability is evaluated separately through the clean and face-swap results in Table 5.

Additional analyses, including optimization-threshold sensitivity, component-level ablations and watermark collision analysis are deferred to the Appendix.

5 Discussion and Limitations

Our method targets transformations that deliberately preserve source identity. This semantic carrier can survive full image resynthesis that destroys spatial and spectral watermarks, but it is not universal. Strong occlusion, severe restoration, or adaptive optimization against the recovery model may remove the signal. Recovery also depends on successful face detection and alignment. An attacker could further attempt to disrupt the watermark by deliberately shifting the identity representation of the protected face. However, under our usage scenario, we assume that the attacker does not have access to the victim's true face embedding. Such manipulation may therefore produce a face-swapped image that neither matches the expected watermark nor maintains high identity similarity to the victim, allowing it to be treated as suspicious rather than simply unverified.

Another limitation is the potential dependence on the identity representation. Many face-swapping systems use ArcFace-like identity encoders, and the observed robustness may therefore partly benefit from shared or similar representations across protection, identity transfer, and watermark recovery. Performance may degrade when an attacker employs a face-swapping method based on a substantially different, unknown identity encoder. Evaluating robustness across more diverse identity representations is therefore an important direction for future work.

6 Conclusion

We study provenance when a protected portrait is used only as an identity reference and contributes no persistent pixels to the generated output. Our source-identity watermarking formulation treats facial identity as the transmission carrier, complementing prior methods that protect target-side image content. Experiments across multiple face-swapping pipelines show that the embedded information remains recoverable after identity transfer while maintaining high visual fidelity and robustness to common image degradations. We hope this setting encourages watermark evaluation under semantic identity transfer and target-free generation, beyond conventional preserving distortions.

References

Seung-hyeok Back, Do Hyun Ki, Juwan Kim, and Seok Bong Yoo. Robust, generalizable proactive face-swapping defense via semantic gradient divergence. In *Proceedings of the Thirty-Fifth International Joint Conference on Artificial Intelligence*, pp. 890–898, 2026.

Tu Bui, Shruti Agarwal, and John Collomosse. Trustmark: Robust watermarking and watermark removal for arbitrary resolution images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025.

Sergio Canu. Face swapping (explained in 8 steps) – OpenCV with Python. Pysource blog tutorial, May 2019. Accessed: 2026-09-26.

Renwang Chen, Xuanhong Chen, Bingbing Ni, and Yanhao Ge. Simswap: An efficient framework for high fidelity face swapping. In *MM '20: The 28th ACM International Conference on Multimedia*, 2020.

Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

FaceFusion. HyperSwap (hyperswap_1a_256): Face swapping model. https://huggingface.co/facefusion/models-3.3.0/blob/main/hyperswap_1a_256.onnx, 2025. Open-source ONNX weights, FaceFusion model release 3.3.0. Accessed: 2026-09-26.

Pierre Fernandez, Guillaume Couairon, Hervé Jégou, Matthijs Douze, and Teddy Furon. The stable signature: Rooting watermarks in latent diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851, 2020.

Gwanghyun Kim, Taesung Kwon, and Jong Chul Ye. Diffusionclip: Text-guided diffusion models for robust image manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2426–2435, 2022.

Marek Kowalski. FaceSwap: 3D face swapping implemented in Python. https://github.com/MarekKowalski/FaceSwap, 2016. GitHub repository. Accessed: 2026-09-26.

Eungi Lee, Seung-hyeok Back, Hyung-Il Kim, and Seok Bong Yoo. Deepprotect: Proactive face-swapping defense using identity blending and attribute distortion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

Junyan Luo, Peipeng Yu, Jianwei Fei, Shiya Zeng, Xiaoyu Zhou, Zhihua Xia, and Xiang Liu. ID-Eraser: Proactive defense against face swapping via identity perturbation. *arXiv preprint arXiv:2604.21465*, 2026.

Reface. Reface: AI face swap app. https://reface.app/, 2020. Commercial application. Accessed: 2026-09-26.

Felix Rosberg, Eren Erdal Aksoy, Fernando Alonso-Fernandez, and Cristofer Englund. Facedancer: Pose- and occlusion-aware high fidelity face swapping, 2022. URL https://arxiv.org/abs/2210.10473.

Tom Sander, Pierre Fernandez, Alain Durmus, Teddy Furon, and Matthijs Douze. Watermark anything with localized messages. In *International Conference on Learning Representations*, 2025.

Matthew Tancik, Ben Mildenhall, and Ren Ng. Stegastamp: Invisible hyperlinks in physical photographs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

VisoMaster Team. VisoMaster: Face swapping and editing toolkit (InStyleSwapper256 model). https://github.com/visomaster/VisoMaster, 2025. GitHub repository. Accessed: 2026-09-26.

Haofan Wang. inswapper: One-click face swapper and restoration. https://github.com/haofanwang/inswapper, 2023. GitHub repository. Accessed: 2026-09-26.

Tianyi Wang, Mengxiao Huang, Harry Cheng, Bin Ma, and Yinglong Wang. Robust identity perceptual watermark against deepfake face swapping. *arXiv preprint arXiv:2311.01357*, 2025a.

Tianyi Wang, Shuaicheng Niu, Harry Cheng, Xiao Zhang, and Yinglong Wang. Nullswap: Proactive identity cloaking against deepfake face swapping. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9945–9954, 2025b.

Yuxin Wen, John Kirchenbauer, Jonas Geiping, and Tom Goldstein. Tree-ring watermarks: Fingerprints for diffusion images that are invisible and robust. In *Advances in Neural Information Processing Systems*, 2023.

Xiaoshuai Wu, Xin Liao, and Bo Ou. Sepmark: Deep separable watermarking for unified source tracing and deepfake detection. In *Proceedings of the 31st ACM International Conference on Multimedia*, 2023.

Ruiyang Xia, Dawei Zhou, Decheng Liu, Lin Yuan, Jie Li, Nannan Wang, and Xinbo Gao. Towards generalized proactive defense against face swapping with contour-hybrid watermark. *arXiv preprint arXiv:2505.19081*, 2025.

Lizhi Xiong, Jun Li, Ziqiang Li, Weiwei Jiang, Zhangjie Fu, and Zhihua Xia. EA-APO: A universal proactive defense against facial manipulation. *IEEE Transactions on Information Forensics and Security*, 21:5414–5426, 2026. doi: 10.1109/TIFS.2026.3696574.

Chao Xu, Jiangning Zhang, Yue Han, Guanzhong Tian, Xianfang Zeng, Ying Tai, Yabiao Wang, Chengjie Wang, and Yong Liu. Designing one unified framework for high-fidelity face reenactment and swapping. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner (eds.), *Computer Vision – ECCV 2022*, pp. 54–71, Cham, 2022. Springer Nature Switzerland. ISBN 978-3-031-19784-0.

Zijin Yang, Kai Zeng, Kejiang Chen, Han Fang, Weiming Zhang, and Nenghai Yu. Gaussian shading: Provable performance-lossless image watermarking for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Xuanyu Zhang, Runyi Li, Jiwen Yu, Youmin Xu, Weiqi Li, and Jian Zhang. Editguard: Versatile image watermarking for tamper localization and copyright protection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Yunming Zhang, Dengpan Ye, Caiyun Xie, Long Tang, Chuanxi Chen, Ziyi Liu, and Jiacheng Deng. Dual defense: Adversarial, traceable, and invisible robust watermarking against face swapping. *arXiv preprint arXiv:2310.16540*, 2023.

Wenliang Zhao, Yongming Rao, Weikang Shi, Zuyan Liu, Jie Zhou, and Jiwen Lu. Diffswap: High-fidelity and controllable face swapping via 3d-aware masked diffusion. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8568–8577, 2023. doi: 10.1109/CVPR52729.2023.00828.

APPENDIX

This appendix provides additional implementation details, analyses, ablations, and qualitative results that complement the main paper. It further examines the training and refinement procedures, sensitivity to design choices, and watermark collision behavior. We also provide additional protected samples, source–target–faceswap examples, and details of the experimental environment.

A RELATED WORK

A.1 ROBUST IMAGE WATERMARKING

Image watermarking embeds recoverable information into visual content for provenance, copyright protection, or authenticity verification. Learned post-hoc methods such as StegaStamp Tancik et al. (2020) and TrustMark Bui et al. (2025) improve robustness to common image degradations and geometric transformations (Tancik et al., 2020; Bui et al., 2025), while localized and tamper-aware methods such as WAM and EditGuard extend watermarking to cropping, splicing, inpainting, and manipulation localization (Sander et al., 2025; Zhang et al., 2024). A separate line integrates watermarking with generative models, for example by modifying a latent diffusion decoder or the initial diffusion noise (Fernandez et al., 2023; Wen et al., 2023; Yang et al., 2024). These methods differ substantially in construction, but their watermark is tied to image content or to a controlled generation process. Our setting instead considers a protected source portrait whose pixels and spatial content need not survive the downstream face swap.

A.2 PROACTIVE WATERMARKING FOR FACE MANIPULATION

Several works apply proactive watermarking specifically to face manipulation. Under the source–target protocol defined in Fig. 1, these methods protect the target/carrier image rather than the identity-providing source considered in our setting. SepMark embeds a watermark into the image to be manipulated and uses robust and semi-robust decoders for source tracing and Deepfake detection (Wu et al., 2023). Dual Defense embeds a robust adversarial watermark into the target face to jointly support traceability and manipulation disruption (Zhang et al., 2023). Identity-perceptual watermarking associates watermark content with facial identity semantics and recovers it after face swapping (Wang et al., 2025a), while CMark places a contour-hybrid watermark in regions designed to remain robust under unknown face-swapping techniques (Xia et al., 2025). These methods address important proactive forensic settings, but the protected image is the target/carrier side of the swap. In contrast, we watermark the source portrait that supplies identity, while the target is selected later by the attacker and is never assumed to be protected.

A.3 SOURCE-SIDE PREVENTION AGAINST FACE SWAPPING

A more closely related line of work protects the identity-providing source itself. NullSwap cloaks source identity features to prevent face-swapping models from recovering the correct identity (Wang et al., 2025b), while DeepProtect combines identity blending and attribute distortion to disrupt identity extraction and degrade downstream face-swapping results (Lee et al., 2026). SGD-Guard similarly targets identity extraction at the source through transferable semantic distortions (Back et al., 2026). ID-Eraser directly perturbs high-level identity representations and reconstructs a natural-looking protected image intended to make the source identity unusable to unknown face-swapping models (Luo et al., 2026). EA-APO instead optimizes transferable adversarial perturbations on source face images to disrupt both face swapping and facial attribute manipulation (Xiong et al., 2026). These approaches share our source-side deployment setting but pursue prevention rather than provenance: their goal is to disrupt or suppress identity transfer.

Prevention, however, does not eliminate the need for provenance. If an attacker observes that a particular face-swapping pipeline fails on a protected source, they may switch to a different or future face-swapping model whose identity extraction mechanism responds differently to the protection. More importantly, they may bypass learned face-swapping pipelines altogether and resort to simpler manipulations, such as cut-and-paste compositing, Photoshop or other cheapfake-style edits. Such

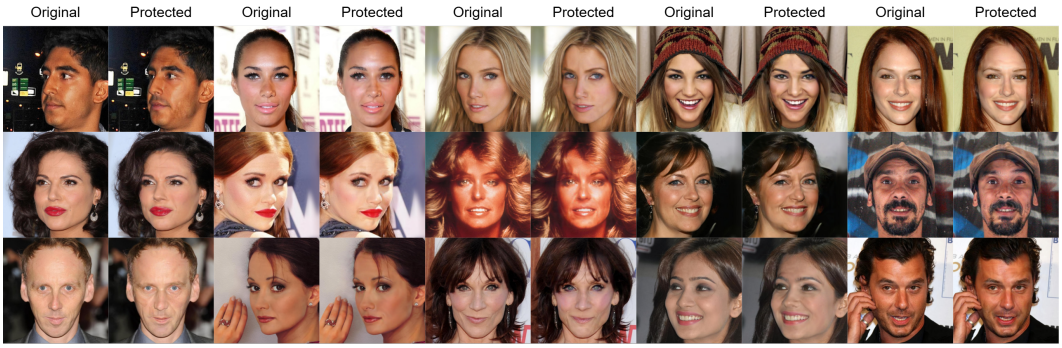

Figure 4: Additional examples of source portraits protected by IdentityMark.

attacks do not necessarily rely on the latent identity representations or optimization pathways targeted by source-side prevention methods, and therefore may not exhibit the intended disruption behavior.

Our goal is complementary: rather than expecting the manipulation itself to fail, we aim to make subsequent misuse traceable. In this sense, provenance provides a fallback layer when prevention is bypassed by a change in attack model or manipulation strategy.

B Hardware and Experimental Environment

All experiments were conducted on an NVIDIA GeForce RTX 3090 GPU with 24 GB of memory. Our implementation uses Python 3.9.25, PyTorch 2.0.1, Torchvision 0.15.2, and CUDA 11.7. The main dependencies include InsightFace 1.0.1 for face processing and identity extraction, LPIPS 0.1.4 for perceptual similarity, and Diffusers 0.30.3. Unless otherwise specified, all training, source-image protection, and evaluation experiments were performed in this environment.

C Additional Qualitative Results

We provide additional qualitative examples to complement the results in the main paper. Figure 4 shows protected source images generated by IdentityMark, illustrating the visual changes introduced during protection across different identities.

Figure 5 further provides qualitative face-swapping examples for several protected source images. For each protected source, we show the face-swapped outputs obtained using multiple target images, demonstrating that the embedded information is evaluated across diverse target identities, poses, expressions, and appearances.

D Training Schedule in Pseudo-Identity Learning

Training uses three phases: the first 10% of epochs train the face branch and shared decoder, the next 60% train only the watermark branch while freezing the face branch and decoder, and the final 30% jointly fine-tune the watermark modules and shared decoder while keeping the face encoder frozen.

E Quality Refinement

The initial protected-image optimization is designed primarily to move the machine-readable face embedding toward the watermark-derived pseudo identity while keeping the rendered portrait perceptually close to the original image. In practice, this stage may introduce image residuals that are not all necessary for maintaining either the pseudo-identity target or the recoverable watermark. We therefore apply a separate quality-refinement stage after the initial protection step. Rather than re-optimizing the diffusion latent, the refinement operates directly in image space and starts from the already protected image.



Figure 5: Additional qualitative examples of face swapping using protected source portraits. Each example shows the protected source and the resulting face-swapped image.

Let I_s denote the original source portrait and $\tilde{I}_s$ the image obtained after the initial identity-realization stage. We parameterize the refined image as

$$I_r = I_s + \mathrm{clip}\left(M \odot (\tilde{I}_s - I_s) + \Delta, -C, C\right), \tag{14}$$

where M is a learned spatial gate that controls how much of the original protection residual is retained, and Δ is a bounded correction residual parameterized as

$$\Delta = \frac{8}{255}\tanh(\Delta_{\mathrm{raw}}). \tag{15}$$

The element-wise clipping budget is defined as

$$C = \min\left(1.1|\tilde{I}_s - I_s| + \frac{2}{255}, 0.16\right), \tag{16}$$

which limits the refinement relative to the original protected-image residual. In our implementation, M is represented at low spatial resolution and bilinearly upsampled, while Δ_{raw} is similarly parameterized at a lower resolution before being added to the image. This parameterization allows the refinement to selectively suppress unnecessary changes instead of freely reconstructing a new portrait.

The refinement jointly optimizes watermark preservation, pseudo-identity preservation, and visual fidelity:

$$\mathcal{L}_{\mathrm{refine}} = \lambda_{\mathrm{id}}\mathcal{L}_{\mathrm{id}} + \lambda_{\mathrm{wm}}\mathcal{L}_{\mathrm{wm}} + \mathcal{L}_{\mathrm{app}} + \lambda_{\Delta}\|\Delta\|_2^2. \tag{17}$$

The identity term is implemented as a hinge penalty that is activated only when the cosine similarity between the refined face embedding and the target pseudo identity falls below the desired identity target,

$$\mathcal{L}_{\mathrm{id}} = \left[\max\left(0,\, \tau_{\mathrm{id}} - \cos\big(E_{\mathrm{id}}(I_r), p_w\big)\right)\right]^2. \tag{18}$$

The watermark term is computed from the frozen recovery network and places greater emphasis on bits that are currently difficult to recover. The appearance term compares the refined image with the original source and combines perceptual similarity, spatially weighted absolute error, pixel-wise squared error, structural consistency, low-frequency chromatic consistency, and image-gradient consistency. The complete objective is optimized jointly at every refinement iteration.

Because reducing image residuals can potentially weaken the embedded identity watermark, we do not simply return the final optimization iterate. Instead, we periodically save candidate images and

independently re-detect and align the face before evaluating each candidate. For every candidate, we measure its cosine similarity to the target pseudo identity, the number of correctly recovered watermark bits, PSNR, LPIPS, and structural similarity relative to the original source. A candidate is considered valid only if it does not degrade the protected anchor in PSNR, LPIPS, or structural quality, while also satisfying a minimum pseudo-identity cosine similarity and a minimum watermark-recovery requirement. Among the valid candidates, we select the one minimizing the combined perceptual and weighted pixel-distance objective; if no refinement candidate satisfies all constraints, the original protected image is retained.

Algorithm 2 summarizes the procedure.

Algorithm 2 Quality Refinement

Require: Original image I_s, protected image $\tilde{I}_s$, target pseudo identity p_w, watermark w, frozen identity encoder E_{id}, frozen recovery network R_w
Ensure: Refined protected image $\hat{I}_s$
Initialize spatial gate M and correction residual Δ
Set $\tilde{I}_s$ as the anchor candidate
for $t = 1, \ldots, T_r$ **do**
    $I_r \leftarrow \mathrm{Render}(I_s, \tilde{I}_s, M, \Delta)$
    $e_r \leftarrow E_{\mathrm{id}}(I_r)$
    $\mathcal{L} \leftarrow \mathcal{L}_{\mathrm{refine}}(I_r, I_s, e_r, p_w, w)$
    Update M and Δ jointly using $\nabla_{M,\Delta}\mathcal{L}$
    Periodically save I_r as a candidate
end for
for each saved candidate I_r **do**
    Re-detect and align its face
    Measure identity cosine, recovered bits, PSNR, LPIPS, and structural quality
    Discard I_r if any validity constraint is violated
end for
$\hat{I}_s \leftarrow$ valid candidate with the best quality objective
if no valid refined candidate exists **then**
    $\hat{I}_s \leftarrow \tilde{I}_s$
end if
return $\hat{I}_s$

F Threshold Sensitivity and Early Stopping

We first study the effect of the identity-similarity stopping threshold used during protected-image optimization. A lower threshold may terminate optimization earlier and preserve image quality, whereas a higher threshold forces the protected-image embedding closer to the pseudo-identity target. We evaluate three thresholds using the same source images, watermarks, and downstream evaluation protocol.

For each setting, we report watermark recovery on the protected source, average recovery under the conventional degradation suite, recovery after SimSwap and InSwapper, and image quality measured by PSNR and SSIM. Degradation BA denotes the average bit accuracy over the fixed set of conventional post-processing attacks used in the robustness evaluation.

Table 6: Sensitivity to the identity-similarity stopping threshold.

Threshold	Actual Cosine	Clean BA	Degradation BA	SimSwap BA	InSwapper BA	PSNR	SSIM
0.85	0.85	99.47	98.11	78.40	81.08	30.27	0.93
0.90	0.86	99.55	98.20	77.38	80.02	31.15	0.94
0.95	0.87	99.55	98.16	75.23	77.38	31.95	0.94

G Contribution of Round-Trip Projection and Quality Refinement

We next isolate the contribution of two stages in the final protection pipeline: the round-trip projection through the learned identity autoencoder and the post-optimization quality-refinement stage. The

former projects the source-conditioned watermark target back onto the learned identity manifold, while the latter suppresses unnecessary image residuals without sacrificing watermark recoverability.

We compare the complete pipeline against variants without round-trip projection, without quality refinement, and without both components. All variants use the same source identities, messages, and downstream face-swapping protocol.

Table 7: Contribution of round-trip projection and quality refinement.

Variant	Clean BA	Degradation BA	SimSwap BA	InSwapper BA	PSNR	SSIM
Full model	99.55	98.16	75.23	77.38	31.95	0.94
w/o round trip	99.36	97.97	75.32	77.47	31.95	0.94
w/o quality refinement	99.63	98.74	78.50	81.52	27.79	0.87
w/o both	99.43	98.68	78.59	81.33	27.76	0.87

H Watermark Collision Analysis

We examine whether the watermark-to-pseudo-identity mapping and the subsequent recovery round trip cause distinct watermarks to collapse toward the same representation. For each of 10 source images, we sample 50 distinct 32-bit watermarks, yielding $\binom{50}{2} = 1{,}225$ watermark pairs per source and 12,250 pairs in total. All statistics are first computed per source and then macro-averaged.

For an input watermark w_1, the watermark autoencoder produces a pseudo-identity p_1. The recovery head predicts a watermark w_2 from p_1, which is decoded again to obtain the round-trip pseudo-identity p_2. We compare pairwise Hamming distances among the watermarks and pairwise cosine similarities among the corresponding pseudo-identities before and after this round trip.

As shown in Table 8, no exact watermark collision is observed: each source retains 50 distinct recovered watermarks, and none of the 12,250 evaluated pairs has zero Hamming distance. The mean pairwise Hamming distance changes only from 15.98 to 15.94 bits, remaining close to the 16-bit expectation for unrelated random 32-bit strings. Similarly, the mean pairwise pseudo-identity cosine similarity changes only from 0.0264 to 0.0292. The per-sample agreement between w_1 and w_2 is 0.9928 ± 0.0160. These results provide no evidence that distinct watermarks collapse to the same recovered watermark or a common round-trip pseudo-identity.

Table 8: Collision analysis before and after the watermark recovery round trip. Each source identity is evaluated with 50 distinct random watermarks (1,225 pairs per source), and values are macro-averaged over 10 source identities.

Metric	Before round trip	After round trip
Unique watermarks per source	50	50
Zero-Hamming pairs	0	0
Mean pairwise Hamming distance (bits)	15.98	15.94
Mean pairwise pseudo cosine similarity	0.026	0.029

I AI Use Statement

Generative AI tools were used to assist with language polishing and to support the development and debugging of code used in the training pipeline. They were not used to formulate the research idea or design the proposed method. Generative AI was also not used to conduct or assess the evaluation; all experimental evaluation and interpretation of the results were performed by the authors. We reviewed all AI-assisted content and take full responsibility for the final manuscript, implementation, and reported results.